BAB II

TINJAUAN PUSTAKA

2.1. Pengenalan Clipper

2.1.1. Pengertian Clipper

Clipper adalah salah satu bahasa pemrograman tingkat tinggi yang termasuk kedalam kelompok x base. Ini merupakan pernyataan yang harus dicamkan sebelum mempelajarinya. Ia bukan lagi sekedar compiler bagi bahasa pemrograman dataBase maupun FoxBase. Bila Clipper mampu mengkompile program yang dibuat dengan kedua bahasa tersebut, itu bukanlah fungsi clipper yang sebenarnya, karena clipper menterjemahkan kembali perintah-perintah kedua bahasa tersebut ke dalam fungsi-fungsi yang hanya dikenal clipper.

2.1.2. Perkembangan Clipper 5.3.

Clipper Compiler pada waktu pemunculan pertamanya hanyalah sekedar program compiler dBase, yaitu untuk mengubah program dBase (berjenis PRG) menjadi program yang berdiri sendiri dalam bahasa mesin (berjenis EXE), sehingga dapat langsung dijalankan dari sistem operasi dan berjalan lebih cepat. Clipper Compiler ini terus berkembang dari versi ke versi terutama terjadi perkembangan yang sangat cepat pada versi winter 85, automn 86 dan summer 87/88 itulah sebabnya mengapa program Clipper Compiler menjadi demikian popular. Banyak programmer yang sangat mengandalkan kemampuan Clipper, karena pada versi

summer 87/88 Clipper bukan lagi hanya sekedar compiler dBase, melainkan sudah menjadi program data base yang berdiri sendiri, bahkan kedalamnya ditambahkan fungsi-fungsi yang sangat didambakan programmer, seperti User Interface yang membuat penampilan lebih baik, menarik serta mudah diatur, programming dalam orientasi tabel, fungsi-fungsi menu dan mengijinkan pemakai mendefinisikan fungsi sendiri. Teknologi yang dibawa Clipper pada waktu itu merupakan hal yang baru dan merupakan lompatan teknologi dibandingkan program data base lain, sehingga membuat Clipper makin banyak dipakai.

Perkembangan selanjutnya adalah pada saat diluncurkannya Clipper Compiler versi 5.0, pada versi ini Clipper sudah mulai menginjak teknologi Object Oriented Programming (OOP) yang memang pada saat itu sedang trend. Konsep OOP tersebut semakin memacu daya guna dan kemampuan Program Clipper, tidak lama kemudian muncul Clipper versi 5.01 sampai versi ini Clipper masih di bawah perusahaan Nantucket. Barulah pada versi 5.2 Clipper diambil alih oleh perusahaan Computer Associated dan diberi nama CA Clipper 5.2.

2.2. Menulis Program

Untuk menulis program Clipper, anda bisa menggunakan program Editor apa saja asal berbasis teks (ASCII), misalnya Side Kick, Edit dari DOS, QE atau editor yang bawaan Clipper sendiri, yaitu Program Editor (PE) dan lain-lain.

Program yang ditulis sebenarnya hanyalah kumpulan teks yang tidak dimengerti oleh komputer. Program yang sedang ditulis tersebut menggunakan extension atau akhiran *.Prg dan kemudian program ini disebut sebagai kode sumber (Source Code). Agar kode program ini dimengerti oleh komputer, maka harus melewati tahap kompilasi dan tahap linking.

2.3. Prosedur Kompilasi dan Linking

Program sumber yang sudah dibuat supaya bisa dilaksanakan oleh komputer harus diubah kedalam bahasa mesin melalui prosedur kompilasi dan linking.

2.3.1. Prosedur Kompilasi

Untuk mengkompilasi program sumber yang sudah dibuat, dapat mengikuti cara-cara sebagai berikut:

Dari Prompt C:\ Ketikan:

C:\ Clipper [Nama File] <enter>

Misal C:\ Clipper Coba1.Prg <enter>

2.3.2. Prosedur Linking

Untuk mendapatkan kode dalam bahasa mesin, object code yang telah terbentuk harus digabungkan dan digandengkan dengan kode mesin yang ada dalam daftar Clipper. Sintax Linking adalah sbb:

C:\ rtlink f1 coba1 <enter>

Setelah proses selesai, anda akan mendapatkan file berjenis EXE (Executable Code).

Dari kedua prosedur linking tersebut, sebenarnya ada cara yang lebih singkat, yaitu dengan cara mengetikkan perintah berikut pada prompt DOS:

C:\ CL Nama File <enter>

Misal C:\ CL Coba1 <enter>

CL adalah nama file batch yang secara otomatis dibuat saat proses instalasi Clipper ke dalam hardisk, jika file ini tidak ada, kita dapat membuatnya sendiri.

2.4. Type Data dalam Clipper

Di dalam Clipper terdapat beberapa jenis data. Adapun jenis data yang dipunyai Clipper adalah:

1. Character

Data *character* digunakan jika data berbentuk string, misalnya nama, alamat, telepon, kota dan lain-lain. Data character dapat menampung null string sampai string berukuran 64 KB.

2. Date

Tipe data *date* digunakan untuk mewakili data tanggal, Clipper bisa mengoperasikan data tanggal 1 Januari 1900 sampai dengan 31 Desember 2999. Clipper juga dapat membuat tanggal kosong.

3. *Logical*

Data Logical yaitu data yang mewakili 2 kondisi, Benar (True) dan Salah (False) dalam Clipper 5.3 data Logical dapat diindeks.

4. *Numeric*

Data *Numeric* yaitu data angka yang dapat dioperasikan secara matematika. CA Clipper 5.3 dapat menampung data numeric dari 10 pangkat -308 sampai 10 pangkat 308 dengan ketelitian dibelakang titik desimal sampai dengan 16 digit.

5. *Memo*

Data *memo* seperti data character, tetepi biasanya untuk data string yang sangat besar dan atau terdiri dari beberapa baris. Jika anda membuat data character pada sebuah file DBF, data tersebut mempunyai panjang sama pada setiap recordnya, tetapi untuk data memo masing-masing record bisa mempunyai panjang data yang berlainan.

6. *Array*

Data *Array* yaitu satu atau lebih variabel yang mempunyai nama yang sama. Jika satu nama tersebut berisi beberapa variabel maka sebagai pembedanya adalah nomor elemen (index/subscript) variabel tersebut.

7. *NIL*

Data *Nil* yaitu untuk menampung sesuatu yang tidak ada isinya. Biasanya dipakai jika anda membutuhkan argumen tetapi tidak mendefinisikan isinya.

8. Object

Dengan data baru ini Clipper sudah mengarahkan dalam OOP. Clipper menggunakan data object berupa data class.

Dari 8 jenis data di atas, yang ada untuk file DBF hanya 5 yaitu character, date, logical, memo dan numeric.

2.5. Operator dalam Clipper

CA Clipper 5.3. mempunyai operator-operator yang dapat dikelompokkan sebagai berikut:

1. Operator Matematik

Digunakan dalam pengolahan data-data yang memeriukan perhitungan secara matematika.

Operator	Kegunaan	untuk jenis data
**,^	Eksponensial	N
*	Perkalian	N
/	Pembagian	N
%	Modulus (sisa pembagian)	N
+	Tambah	N,D
-	Kurang	N
-	Sambung	C
++	Bertambah (naik)	N,D
--	Berkurang (turun)	N,D

Keterangan : C = Karakter

D = Date

N = Numeric

2. *Operator Relasi*

Digunakan untuk membandingkan 2 buah data dan menghasilkan data logika, True (.T.) berarti Benar atau False (.F.) berarti Salah.

operator	Kondisi	untuk jenis data
<>!=#	Tidak sama	C,D,L,N,M,NIL
<	Lebih kecil	C,D,L,N,M,A,O,NIL
>	Lebih besar	C,D,L,N,M
<=	Lebih kecil / sama dengan	C,D,L,N,M
=	Sama dengan	C,D,L,N,M,NIL
>=	Lebih besar / sama dengan	C,D,L,N,M
$	Ada dalam string	C,M

Keterangan: C = Karakter O = Object M = Memo

N = Numeric D = Date NIL = Nil

A = Array L = Logic

3. *Operator Logika*

Digunakan untuk menggabungkan beberapa ekspresi logika dan akan menghasilkan sebuah ekspresi logika dari beberapa ekspresi logika tersebut.

a. .And. **ekspresi Dan**

b. .Or. **ekspresi Atau**

c. .Not. **ekspresi Tidak**

2.6. Statement Kendali

Digunakan untuk memilih bagian program yang akan dikerjakan sesuai dengan kondisi yang diberikan. Ada 2 buah statement kendali yang ada dalam Clipper yaitu Statement If dan Statement Do Case.

2.6.1. Statement If

Statement If akan mencek suatu kondisi dan menentukan apakah kondisi tersebut benar atau salah, kemudian melakukan suatu kegiatan sesuai dengan nilai kondisi tersebut.

Struktur dari Statement If dapat berupa:

a. **If ... endif**

b. **If ... else ... endif**

c. **If ... elseif ... else ... endif**

a. Struktur If ... endif

```
If ungkapan
    statement
endif
```

b. Struktur If ... elseif ... endif

```
If ungkapan
    statement 1
else
    statement 2
```

Endif

c. ***Struktur If ... elseif ... else ... endif***

If ungkapan 1

statement 1

elseif ungkapan 2

statement 2

elseif ungkapan N

statement N

else

statement terakhir

Endif

Keterangan:

Ungkapan adalah kondisi yang akan diseleksi oleh statement if, sampai kondisi yang diseleksi terpenuhi kemudian diproses.

2.6.2. Statement Do Case ... Case ... Otherwise ... Endcase

Bentuk dari struktur *if ... elseif ... else ... endif* yang kompleks seperti di atas sebenarnya dapat disusun dengan cara yang lebih sederhana yaitu dengan menggunakan statement ***Do ... Case.***

Statement *do ... case* memiliki aturan penulisan sebagai berikut:

Do Case

Case <kondisi 1>

Statement - 1 (hasil-1)

Case >kondisi 2>

Statement - 2

Case <kondisi N>

Statement – N

Otherwise

Statement terakhir

Endcase